Forensic Science International: Genetics 5 (2011) 126–132

Contents lists available at ScienceDirect

Forensic Science International: Genetics

journal homepage: www.elsevier.com/locate/fsig

SAM: String-based sequence search algorithm for mitochondrial DNA database queries

Alexander Röck [a], Jodi Irwin [b], Arne Dür [a], Thomas Parsons [c], Walther Parson [d,*]

[a] Institute of Mathematics, University of Innsbruck, Technikerstrasse 13, 6020 Innsbruck, Austria
[b] The Armed Forces DNA Identification Laboratory, 1413 Research Blvd., Rockville, MD 20850, USA
[c] The International Commission on Missing Persons, Alipašina 45 A, 71000 Sarajevo, Bosnia and Herzegovina
[d] Institute of Legal Medicine, Innsbruck Medical University, Müllerstrasse 44, 6020 Innsbruck, Austria

ARTICLE INFO

Keywords:
mtDNA databases
Phylogenetic
Alignment
Sequences
EMPOP

ABSTRACT

The analysis of the haploid mitochondrial (mt) genome has numerous applications in forensic and population genetics, as well as in disease studies. Although mtDNA haplotypes are usually determined by sequencing, they are rarely reported as a nucleotide string. Traditionally they are presented in a difference-coded position-based format relative to the corrected version of the first sequenced mtDNA. This convention requires recommendations for standardized sequence alignment that is known to vary between scientific disciplines, even between laboratories. As a consequence, database searches that are vital for the interpretation of mtDNA data can suffer from biased results when query and database haplotypes are annotated differently. In the forensic context that would usually lead to underestimation of the absolute and relative frequencies. To address this issue we introduce SAM, a string-based search algorithm that converts query and database sequences to position-free nucleotide strings and thus eliminates the possibility that identical sequences will be missed in a database query. The mere application of a BLAST algorithm would not be a sufficient remedy as it uses a heuristic approach and does not address properties specific to mtDNA, such as phylogenetically stable but also rapidly evolving insertion and deletion events. The software presented here provides additional flexibility to incorporate phylogenetic data, site-specific mutation rates, and other biologically relevant information that would refine the interpretation of mitochondrial DNA data. The manuscript is accompanied by freeware and example data sets that can be used to evaluate the new software (http://stringvalidation.org).

1. Introduction

In forensic science mitochondrial DNA (mtDNA) haplotypes are primarily generated by direct sequence analysis of PCR amplified fragments that result in overlapping strings of consecutive nucleotides and build a final consensus sequence of multiple reactions. When formulating guidelines for the reporting of mtDNA data the commission of the International Society for Forensic Genetics [1] continued the general scientific tradition of sequence annotation in a difference-coded position-based format relative to the corrected version (rCRS, revised Cambridge Reference Sequence [2]) of the first sequenced mtDNA (CRS [3]). Forensic nomenclature rules were later confirmed and extended by guidelines issued by the European DNA Profiling Group [4]. Still, some sequences have been observed for which the application of these rules did not result in a single unambiguous annotation, but instead allowed for multiple plausible alignments (haplotypes). This was typically encountered in sequence stretches of identical nucleotides—so-called homopolymeric tracts, where the exact position of an insertion or deletion (indel) could not be determined. This poses a general problem for mtDNA database searches since different annotations between the query and the database haplotypes may lead to search results that do not include identical or nearly identical sequences. Attempts have been made to solve complications that arise when novel length variation is encountered by following an operational approach [5]. However, this strategy has never been fully adopted by the forensic community as it introduced "jumping alignments" that occasionally put haplotypes at further distances from each other than the actual mutations would suggest [6]. Therefore, a phylogenetic approach to mtDNA sequence alignment has been formulated [6] that has proven useful and thus forms the basis for haplotype annotation in the EMPOP database (http://www.empop.org). Here alignment and nomenclature is based on the phylogeny of mtDNA, where mutational events are inferred through comparison to the nearest (known) evolutionarily related sequences. This approach

* Corresponding author. Tel.: +43 512 9003 70640; fax: +43 512 9003 73640.
E-mail address: walther.parson@i-med.ac.at (W. Parson).

1872-4973
doi:10.1016/j.fsigen.2010.10.006

has some appealing characteristics. Firstly, it permits nomenclature based on scientific inference of the case in hand rather than an arbitrary rule-based approach. Secondly, by reflecting the evolutionary history of the mtDNA molecule, the phylogenetic approach permits an indication of the true genetic distance between the sequence and its nearest relatives. Thirdly, and most importantly, the phylogenetic approach is in concert with the convention of mtDNA annotation in related fields such as population genetics and medical genetics, between which data are shared and compared with the forensic community on a frequent basis.

A disadvantage of the methodology proposed in [6] with respect to database searching by a forensic practitioner, particularly in relation to large shared databases with many users, is the requirement that users would need to be cognizant of the details of the phylogenetic annotation used in the database. Another disadvantage is that the nomenclature for a particular sequence motif may change over time, as additional phylogenetic insight is gained concerning the evolutionary history of the mutation events in question. This requires a high degree of cognizance for the user, although the number of affected sequences may be very small.

Given these complications we explore in this paper the performance of a modified mtDNA database search engine that is based on the comparison of strings of bases that comprise the mtDNA sequences rather than the conventional haplotypes that are recorded as differences to the rCRS. Sequence comparisons then become independent of nomenclature. If haplotypes (strings) consistent with the queried sample exist in the reference database, nomenclature differences will not confound the retrieval of these matches. Moreover, this type of paradigm may provide additional flexibility to incorporate phylogenetic data, site-specific mutation rates, and other biologically relevant information that would refine the interpretation of mtDNA evidence. Under this model, the database user would not be required to have a precise knowledge of the phylogeny used in the database. This paradigm not only removes the notation subjectivity on the user's part and standardizes database searches, but is also designed very specifically to retrieve exact matches, while incorporating mtDNA specific characteristics (unlike other available searches, such as BLAST). SAM is intended to replace conventional mtDNA database searches that compare difference-coded haplotypes and, as a result, ensures that query haplotypes will be captured in a database search regardless of alignment and annotation. Ultimately, this string-based search engine offers to the forensic community a method by which reliable mtDNA database searches may be performed.

2. Methods

2.1. String-based search

SAM, the string-based search software presented herein, is freely and anonymously accessible via the EMPOP internet database ([7]; www.empop.org; version 2.1). According to EMPOP's terms of reference, stored haplotypes are not presented in a downloadable format. However, in order to provide the user with an open set of mtDNA haplotypes, an example dataset is also freely downloadable to evaluate the results of the string-based query (see below for further information on the validation of the software). The engine uses bounded-distance dynamic programming ([8] pp. 263–265, [9]) to find all database profiles that differ as strings from the query profile by up to 5 differences. This value has been selected as this has been practice in forensic database queries. The user can select the maximum number (d) of differences from the range 0–5, and choose between literal or pattern match mode to specify how symbol differences between query and database strings are counted. The EMPOP database uses the IUPAC code to designate sets of nucleic acid bases. In literal match mode only equal symbols are matched. In pattern match mode two symbols match if one set is contained in the other set. For example, setting the sequence range to the single position 152 and searching for 152Y in literal match mode with zero differences yields all those database strings that harbor this exact mixture (e.g., in case of heteroplasmy). Switching to pattern match mode using the same query settings yields all database strings of EMPOP, since no symbols other than Y = {C,T}, C, and T are found at this position in the database.

2.2. Accounting for hot spots

Length variants that are known hot spots for indels can be ignored by the user. In the current version this involves the C-runs around positions 16193, 309, and 573, and the T-run around position 455 relative to the rCRS as these constitute the four major fast evolving insertion/deletion events that are also known to vary between tissues of an individual. Therefore one may choose to exclude this variability from a database search. Further indel regions can be added to this list later, if necessary, e.g., indels around position 965 for coding region sequences as suggested in Table 4 of [10]. The user can select these hot spots individually in order to exclude such variants from a search. To optimize the run time of a search, the number of differences plus the number of ignored indels has been limited to $d + 6$ per queried region. This is sufficient for the vast majority of mtDNA haplotypes. In the rare event of haplotypes that require a higher number of differences, the size of a region can be reduced to fully take advantage of the software. Here a region refers to a contiguous range of positions relative to the rCRS, e.g., HVS-I (16024-16365), HVS-II (70-340), control region (16024-576), or single positions in the coding region as used in RFLP analyses (note that coding region data are yet stored in EMPOP but will only become available when their number is high enough to provide useful results). The user is free to select any region within the control region of a query and is not limited in the number of searched regions. The small values for the maximum number of differences (5) and the addition for ignored indels (6) are necessitated by the fact that a query haplotype is not compared to the rCRS alone but to all database profiles (more than 10,000 in EMPOP 2) representing a major part of (forensically relevant) world-wide mtDNA lineages. See example 1 below for the effect of ignoring indels.

2.3. Three-step procedure of the string-based search

Given a query string, either represented as differences to the rCRS or as a string of bases, the execution of a string-based database search by SAM initiates the consecutive operation of three steps, in which

(1) the neighboring database strings are determined which include all database strings (with identical or smaller sequence range) that differ from the query string by at most d differences and where the sum of the number of differences and the number of ignored indels does not exceed $d + 6$,
(2) for every neighboring database string, one most parsimonious edit transcript (defined below) is calculated, and
(3) for each neighboring database string selected interactively by the user, one maximum likelihood edit transcript is computed.

An edit transcript specifies by means of substitutions and indels how to transform the database string into the query string ([8] pp. 215–217), and may also contain ignored indels. For instance, the transcript G16129A -16187.1T C309.2- means that G at position 16129 in the database string is replaced by A, that T is inserted in

the database string after position 16187, and that C at position 309.2 in the database string is deleted. For split sequences, i.e. query profiles with a sequence range composed of several regions (e.g., HVS-I and HVS-II), the algorithm treats each region separately and then combines the results. The profiles of the EMPOP database are stored in position-based format relative to the rCRS following the rules proposed in [6] for phylogenetic alignment. However, the transcript from a database string to a query string computed by the algorithm is independent of the position-based format of the query string. Note that all positions used in the transcript refer to the positions of the database string which is supposed to be correctly aligned. In the database string, the runs at the hot spot positions 16193, 309, 455, and 573 are located by checking the string at these positions for the repeating symbol C or T and by scanning the string to the left and to the right for repetitions. E.g., the string described by 16183C 16188T 16189C 16193.1C relative to the rCRS has a run from position 16189 to position 16193.1, and all C-indels in this run are ignored if the user has decided to ignore the hot spot at position 16193. Combining the difference-coded annotation of the database string relative to the rCRS with the short transcript from the nearest database string to the test string yields a difference-coded annotation of the test string with respect to the rCRS. E.g., searching the CR profile CHN.ASN.000451 from [6] (haplogroup B4a1a1a, see below) in EMPOP 2 with default options results in three neighbors at distance 2. One of these base strings is separated only by length variants at hot spot 309 and a deletion in the AC-repeat at position 524 (C308- C309- C309.1- A523- C524-). Phylogenetic alignment of the query string is derived from the alignment of the base profile of the EMPOP database and the transcript (Table 2).

2.4. Determination of neighboring base profiles

In the first step of the string-based database search, mtDNA profiles whose sequence ranges cover that of the query profile are converted into strings and shortened to represent the ranges of the query profile, if necessary. The length n of the resulting string of a full CR profile is about 1122 nucleotides (depending on the number of indels with respect to the rCRS). The number of differences between each database string and the query string is computed and database profiles exceeding d differences are excluded from the search. In general the number of differences can be obtained by filling in the dynamic programming table with n rows and m columns, where m is the length of the database string ([8] pp. 224–225). Here operation weight zero is used for ignored indels and weight one for all other operations.

The programming table has about 1122 times $1122 \sim 1.26$ million entries but can be reduced significantly if the sum of number of differences and the number of ignored indels is bounded. In EMPOP 2 a suitable bound for the sum is $d + 6$ where $d <$ =5. By filling in the reduced dynamic programming table with n rows and $(d + 7)$ diagonals and checking that the last entry does not exceed d, the neighboring base profiles are determined ([8] pp. 263–265). Here we extend the assertion of Gusfield, p. 264, that the size of the strip in the table can be reduced by half, if the lengths of the strings are equal, to the general case where the lengths may differ. In the typical case $n = 1122$ and $d = 5$ the reduced dynamic programming table has only 13464 entries, which is about 1% of the full table.

2.5. Log odds

In the second step of the string-based database search, for every neighboring base string one most parsimonious (MP) edit transcript to the query string is calculated, i.e. one transcript with the minimum number of differences. To prefer transcripts with well-known differences we apply a weighting of the differences by log odds derived from a population dataset of 7074 CR profiles from EMPOP. In addition to the weights associated to point mutations, weights for length changes of the C-runs around positions 16193, 309, and 573, the T-run around position 455, and the AC repeat around position 524 are also used. The total weight of a transcript is the sum of the weights of the differences, and among all MP transcripts a transcript with minimum total weight is computed. E.g., for the rCRS as the database string, the transcript A523- C524- (0.34) is put in favor of the transcripts C522- (3.0) A523- (3.0) or A523- (3.0) C525- (3.0) where the log odds are shown in the parentheses following the point mutations or the length changes. For the same reasons the transcript -524.1A -524.2C (0.39) is preferred to the transcripts -524.1A (3.0) -525.1C (3.0) or -523.1C (3.0) -523.2A (3.0).

The third step of the database search is optional. For a database profile selected by the user the MP condition for the transcript is relaxed and at most 5 additional differences are tolerated. Then a transcript with minimum weight is called a maximum likelihood (ML) transcript because the weights are log odds, and can be enumerated similarly to MP transcripts ([8] pp. 321–325). In most cases the MP transcript of the second step is also ML but there are exceptions caused by tandem repeats or if the mitochondrial phylogeny suggests otherwise as discussed in example 2 below.

2.6. Regular edit transcripts

While the computation of the edit distances is fast, the computation of the MP or ML transcript is much more time consuming because ignoring indels at some hot spots causes huge numbers of MP or relaxed transcripts with at most 5 additional differences. For instance, if indels at the hot spots 16193, 309, 455, and 573 are ignored and 5 is chosen for the maximum number of differences, there are more than 6 billion MP transcripts from the CR-profile AF016 from [11] to the CR-profile CHN.ASN.000451 from Table 2. Hence we restrict the possible transcripts to a subset that can be enumerated faster but is sufficient for forensic applications.

We define an edit transcript as "regular" if the following three conditions on indels are satisfied. Here a symbol equates to one of the IUPAC codes for nucleotides, a block is either a single symbol or several subsequent symbols, and prefixes or suffixes of blocks have at least one symbol. A prefix of a block is called repeating if the whole block can be obtained by repeating the prefix. Block insertions are obtained by combining all symbol insertions at subsequent positions, and block deletions are obtained by combining all symbol deletions at subsequent positions.

(1) If a block deletion is immediately followed 3′ by a block insertion, no suffix of the deleted block coincides with a prefix of the inserted block.
(2) If a block insertion is immediately followed 3′ by a block deletion, no suffix of the inserted block coincides with a prefix of the deleted block.
(3) No block is inserted or deleted 5′ to a match of the shortest repeating prefix of the block.

Roughly speaking conditions (1) and (2) prohibit subsequent block indels that insert and delete corresponding symbols, and can be fulfilled by canceling out the corresponding symbols. E.g., for the rCRS, the transcript C308- C309- -309.1C can be reduced to C309-, and the transcript A523- C524- -524.1A -524.2C can be reduced to the empty transcript. Condition (3) enforces the 3′ notation for indels and can be satisfied by inserting or deleting symbols at the end of the run. E.g., for the rCRS, the transcript -42.1C is transformed to -44.1C as positions 42 through 44 display C-residues in the rCRS (Table 3).

More generally, the insertion or deletion of CCC 5′ to a matching C would violate condition (3). The same applies to an insertion or deletion of ACAC 5′ to a match of AC. Thus, the insertion of ACAC with respect to the rCRS 3′ of position 524 is regular while a single insertion of C 3′ to position 524 would be irregular as position 525 also displays a C (Table 3). For instance, if all indels at the hot spots 16193, 309, 455, and 573 are ignored and 5 is chosen for the maximum number of differences, there are only three regular MP transcripts from the CR-profile AF016 to the CR-profile CHN.ASN.000451 from Table 2, namely [C308- C309- C309.1-] A523- C524-, [C308- C309- C309.1-] A523- C525-, and [C308- C309- C309.1-] C522- A523-.

The symbolwise traceback method of dynamic programming generates transcripts by scanning the strings from right to left and recursively following branches for deletion, insertion, and substitution or match until both strings are exhausted ([8] pp. 221–223). This recursion can be adapted to efficiently generate only regular transcripts by keeping track of the current block indel and the last block indel or block match. The adapted recursion avoids branches where conditions (1) or (2) are violated by checking that, for opposed block indels, the current block is not a prefix of the last block and returning otherwise. To satisfy condition (3) the two branches for insertion or deletion are refined into four branches for regular or temporarily irregular insertion or deletion. In the branches for temporarily irregular insertion or deletion, the adapted recursion either continues until the block insertion or deletion becomes regular, or returns. E.g., for base string rCRS, test string 524.1A 524.2C, and current transcript -524.1A -524.2C, the insertion -524.1C is irregular but the block insertion -524.1A -524.2C is regular, and the recursion continues. On the contrary, for base string rCRS, test string 16193.1C 16193.2C, and current transcript -16189.1C -16189.2C, both the insertion -16189.1C and the block insertion -16189.1C -16189.2C are irregular and the recursion returns. As the branches for temporarily irregular insertions or deletions exhibit no subbranching, the computational overhead for the two new branches is small. We traverse the five branches in the order indels first, followed by substitutions or matches. This puts indels or substitutions as close as possible to the 3′ end since the algorithm updates the optimal transcript only if necessary.

3. Results and discussion

Insertions and deletions in length variants can be aligned in multiple ways when they are reported as differences to the rCRS, simply because in many cases it is impossible to know the position where the mutation event occurred. This may lead to ambiguity in database searches if sequences are stored in a different annotation than the queried sequence is formulated. To avoid an adverse effect of mtDNA annotation on database search results we propose SAM, a string-based search that is immune to different versions of mtDNA sequence annotation, as long as they are a valid translation of the consecutive nucleotide string. This was achieved by introducing an algorithm that converts difference-coded query haplotypes to position-free nucleotide strings, which are then compared to the database sequences that were also presented in string format.

While perfectly consistent haplotypes are always properly returned under the string paradigm, a custom alignment algorithm is needed that reflects our understanding of mtDNA control region variation, mutation and evolution, in order to report the results of the search when sequences differing at one or more positions are included. Our model offers the option to ignore hot spot indels around positions 16193, 309, 455, and 573 (this list can be extended if necessary) to reduce the effect that mismatches at these positions would have on the database hits. These hot spots can be targeted individually depending on the purpose of the search. A phylogenetic query would exclude all hot spots as they have very little or no relevance for the result, whereas a typical forensic search of the database may include some indels while ignoring others. This prevents closely related or even identical sequences that differ only in hot spot indels from being either overlooked or put at further distance in a database query. Example 1 below highlights the power of the option to exclude length variants, to find the nearest matching neighbors in the database and thus avoid search result distances that are distorted by highly variable positions. It is known that indels around positions 16193, 309 and 573 have a high evolutionary rate in a variety of haplogroups and are therefore considered mutational hot spots at the population level. It has also been established that indels around positions 16193 and 309 have an elevated mutation rate in family pedigrees [12] and even show variation between tissues of a single individual. Only few data are available on the individual mutation rate around position 573 and more research is needed to evaluate this and other length variant regions to determine how these should be treated in forensic database comparisons.

Example 1: We use the EMPOP string-based query of the B4a1a1a CR sequence CHN.ASN.000451 with the annotation of [5,13] 16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 310DEL 523DEL 524DEL from [6] to highlight

Table 1
Search results for querying profile CHN.ASN.000451 with range 16024-16365 73-340 in EMPOP 2.

Search result without ignoring indels		
Number of differences to profile CHN.ASN.000451	Number of haplotypes	Haplogroups of haplotypes
0	0	
1	0	
2	0	
3	1	B4a1a1a
4	5	B4
5	4	B4, B4a1a
6+	10889	
Search result ignoring indels at 309		
Number of differences to profile CHN.ASN.000451	Number of haplotypes	Haplogroups of haplotypes
0	1	B4a1a1a
1	0	
2	4	B4, B4a1a
3	19	
4	37	
5	56	
6+	10782	

the importance of taking hot spot indels into account. To incorporate as much data as possible the query region is set to 16024-16365 73-340 thus including all 10899 HVS-I and II database profiles of EMPOP 2. A query considering all indels as individual differences results in no exact match. The nearest neighbor is found to be the B4a1a1a sequence AF016 from [11] at distance 3 with differences C308- C309- C309.1-. Ignoring indels at hot spot 309, which is the default configuration in EMPOP 2, puts the same database profile at distance 0 since all differences are deletions in the C-stretch 5′ to position 310 compared to the rCRS (Table 1).

3.1. Validation

The new string-based search method guarantees that base profiles identical to the query haplotype are found in the course of a database search regardless of the annotation of the haplotypes. As a proof of concept one would expect that none of the database haplotypes are missed in a search independent of their annotation. We confirmed this general concept by querying, in nucleotide string format, 2345 individual difference-coded haplotypes stored in a database and retrieving all corresponding pairs of sequences that matched. Next, we queried 10 different annotations of the B4a1a1a profile CHN.ASN.000451 (Table 2) in EMPOP 2 and verified that the different annotations always resulted in the same output. Finally, we showed that newly observed haplotypes not yet stored in the database do not lead to exact matches when queried against the database. This was shown by example 1, where the nearest neighbor was found at distance 3 when counting all indels as individual differences.

Additionally we provide a subset of the EMPOP 2 database that may be used to experiment with and test the software. The test suite is accessible via http://stringvalidation.org and is a mirror of the actual EMPOP 2 database as far as the layout and string-search engine is concerned. This test version is freely accessible without registration and the haplotypes are made available for the user. The database contains three populations [14–16] that can be downloaded in difference-coded format in the download section of the website.

3.2. Comparing string-based and position-based search

In order to highlight the benefits of the new search engine, we present an example where the two query models produce strikingly different search results.

Example 2: The homopolymeric C tracts around positions 16189 and 310 can be challenging when annotating sequences in difference-coded format. Besides these stretches, there is significant variation in the region around position 60, which harbors a short T-stretch and several transitions and transversions. While motif 55C 57C represents the standard motif for haplogroup (hg) H15 [17], additional transitions and insertions cause this motif to degenerate when annotated most parsimoniously. The CR profile CN253 from [AFDIL 2006, unpublished] (55C 56T 57C 60.1T 93G 263G 309.1C 315.1C 573.1C 573.2C) harbors the hg H15 motif but could also be annotated as 54.1C 56C 93G 263G 309.1C 315.1C 573.1C 573.2C, most parsimoniously. This annotation would destroy the specific pattern 55C 57C and would therefore mistakenly suggest to assign this haplotype more generally somewhere within haplogroup R0 (in which H15 is nested). A search in EMPOP 2 using the new string search is independent of the annotation used and always produces the full match (haplotype contained in EMPOP 2.1). A position-based search would yield disparate results depending on the nomenclature used (Table 4). Both queries were carried out using default values given by the web interface. Regarding the output of the search and

Table 2
Contrasting phylogenetic and operational nomenclature of profile CHN.ASN.000451. Phylogenetic alignment of profile CHN.ASN.000451 is gained taking into consideration the phylogenetic nomenclature of profile AF016 from [12] that is stored in the EMPOP database.

Sample information	Haplogroup	Phylogenetic nomenclature	Operational nomenclature
AF016	B4a1a1a	16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 309.1C 315.1C	
CHN.ASN.000451	B4a1a1a	16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308DEL 309DEL 315.1C 523DEL 524DEL	16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 310DEL 523DEL 524DEL
10 different annotations for profile CHN.ASN.000451			
16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309- 315.1C 523- 524-			
16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309T 310C 523- 524-			
16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 309- 310C 523- 524-			
16182C 16183- 16189C 16193.1C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309- 315.1C 523- 524-			
16182C 16183C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 310C 315- 523- 524-			
16182C 16183- 16189C 16193.1C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309T 310C 523- 524-			
16182C 16183- 16188.1C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309- 315.1C 523- 524-			
16182C 16183- 16189C 16193.1C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 309- 310C 523- 524-			
16182C 16183- 16188.1C 16189C 16217C 16247G 16261T 16519C 73G 146C 263G 308- 309T 310C 523- 524-			
16182C 16183- 16189C 16193.1C 16217C 16247G 16261T 16519C 73G 146C 263G 308T 310C 315- 523- 524-			

Table 3
Contrasting regular and irregular indels. Insertions of all four bases have been described at position 42. The insertion of a C (42.1C) is an irregular term as this position is followed by two C-residues in rCRS. To comply with regularity it is notated at the 3′ end of the C-tract (44.1C). Similar applies to the deletion of block AC at positions 515 and 516. This is transformed to 523- 524-. Bases in bold face denote runs.

	3	4	4	4	4	4	4	4	4	4	4			5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
	9	0	1	2	2	3	4	4	5	6	7			0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	
					.			.																											
					1			1																											
5′												3′	5′																					3′
rCRS	C	T	C	T	-	**C**	**C**	-	A	T	G		rCRS	C	C	A	G	C	**A**	**C**	**A**	**C**	**A**	**C**	**A**	**C**	**A**	**C**	C	G	C	T	G	
42.1C	C	T	C	T	C	C	C	-	A	T	G		515-	C	C	A	G	C	-	-	A	C	A	C	A	C	A	C	C	G	C	T	G	
													516-																						
44.1C	C	T	C	T	-	C	C	C	A	T	G		523-	C	C	A	G	C	A	C	A	C	A	C	A	C	-	-	C	G	C	T	G	
													524-																						

Table 4
Query results obtained by position-based and string-based search of both phylogenetic (55C 56T 57C 60.1T 93G 263G 309.1C 315.1C 573.1C 573.2C) and operational (54.1C 56C 93G 263G 309.1C 315.1C 573.1C 573.2C) nomenclature of CR haplotype CN253.

Number of differences to CN253	Number of haplotypes			Haplogroups of haplotypes
	Position-based search		String-based search	
	Phylogenetic alignment	Rule-based alignment		
0	1	0	1	H15
1	0	0	0	
2	2	0	2	H15
3	6	31	37	R0, H15, H15a1
4	6	247	252	
5	36	424	426	
6+	7279	6628	6612	

without additional information it is not *a priori* clear whether this profile should be annotated as a representative of haplogroups H15 or R0. While the nearest neighbors with 2 differences belong to hg H15, the majority of neighbors with 3 differences belong to haplogroup R0. The profiles of hg R0 are separated by differences -54.1C (3.00) A56C (3.00) from the query profile. The high weight of the differences suggests however that the given MP transcript is not equal to the ML transcript. The ML transcript, which can be calculated on request in the detailed output view of a sample (in EMPOP 2) is T55C (1.05) A56T (1.18) -56.1C (1.12), thus generating less cost. In this example we typed the hg H15 informative SNP 6253 after the analysis and confirmed the hg H15 status of this profile.

3.3. Directions for use within EMPOP

SAM is primarily designed for use within the EMPOP database (http://www.empop.org). The interface for the string-based search provides default values where reasonable thereby minimizing the need for additional action by the user. The software allows for 2 different input modes of the query profile: 1) as differences to the rCRS or 2) as a sequence string, i.e. a consecutive string of nucleotides. A query profile is made up of one or more ranges (e.g., 16024-576 for a full CR profile, 16024-16365 73-340 for a split profile with HVS-I and HVS-II) with the appropriate differences or sequence strings respectively. Each of these ranges has to be entered separately into the input mask. As explained above, the user can choose between pattern and literal match and opt for positions where indels in length variants are to be ignored (standard setting is ignoring all hot spot indels). Furthermore, the number of differences between the query profile and the database profiles can be selected between 0 and 5 (default value 5).

The output of a search summarizes the input values and presents a table with the number of (cumulative) haplotypes found in the database, sorted by the number of differences to the query profile. Registered users have the possibility to view the individual haplotypes and differences to the query profile in a separate tab. In addition, the detail view of a database profile also provides the option to calculate the ML transcript on demand.

The complete phylogenetic revision of 5,173 mtDNA haplotypes stored in EMPOP (http://www.empop.org) release 1 (2006-2009) brought only 10 annotation changes due to newly observed lineages and 2 annotation changes due to erroneously introduced variants in release 1, as we did not use automated software checks but manual control at that time. All changes are recorded in the release history since the launch of EMPOP 2.

4. Conclusion and forensic relevance

SAM is a valuable contribution for mtDNA database queries as it takes the burden of sequence alignment from the user. The string-based algorithm is immune against different versions of sequence annotation and guarantees that an identical sequence to the query is found in the database. This is in contrast to BLAST, which is a heuristic approach that may miss the relevant haplotype. SAM avoids a disadvantageous effect on the query result that may otherwise occur by underestimating the frequency of forensic mtDNA sequences.

The string-based search algorithm described herein is customized for the user to compare and evaluate the results generated by a difference-coded phylogenetic database query. This tool can discover database sequences that may have been missed in a position-based search due to annotation differences between query and database sequences. It can be viewed as a safe-guard for sequence searches where the practitioner is in doubt about multiple possible sequence annotations. The string-based algorithm primarily aims at finding identical sequences (including the option to disregard differences in hot spot regions), and is intended to replace the phylogenetic-based search option that has been the standard EMPOP query option (and is still for other mtDNA databases). Also, it is important to understand that the number of neighbors of a search may vary slightly between position-based (phylogenetic) and string-based search results, since the former is dependent on the rCRS as an intermediate for interpretation, whereas the latter is based on maximum parsimony between the query sequence and the database sequence only. That however, does not hamper the use of the string-based search tool; in contrast, it adds further information to the search result.

Currently EMPOP is designed to hold forensically relevant data, which mostly comprises HVS-I, HVS-II and control region sequences. Nevertheless, the algorithm presented herewith is also capable of processing full mitochondrial genomes, and has been tested with a database of 3217 full mitochondrial genomes from the literature (data not shown). While current runtime is adequate if the number of ignored differences per region is limited to five, the time used for exhaustive search of optimal transcripts increases rapidly with higher values. We therefore follow the approach of calculating MP and ML transcript separately. For the initial query output MP transcripts only are presented making the search significantly faster. The computation of the ML transcript for a specific neighboring base profile is optional for the user.

Acknowledgments

The authors would like to thank Martin Bodner, Liane Fendt, Sabine Lutz-Bonengel, Lourdes Prieto Solla, Kim Sturk and Bettina Zimmermann for helpful discussion. The study was supported by the FWF Austrian Science Fund (TR397).

References

[1] W. Bär, B. Brinkmann, B. Budowle, A. Carracedo, P. Gill, M. Holland, P.J. Lincoln, W. Mayr, N. Morling, B. Olaisen, P.M. Schneider, G. Tully, M. Wilson, DNA Commission of the International Society for Forensic Genetics: guidelines for mitochondrial DNA typing, Int. J. Legal Med. 113 (2000) 193–196.
[2] R.M. Andrews, I. Kubacka, P.F. Chinnery, R.N. Lightowlers, D.M. Turnbull, N. Howell, Reanalysis and revision of the Cambridge reference sequence for human mitochondrial DNA, Nat. Genet. 23 (1999) 147.
[3] S. Anderson, A.T. Bankier, B.G. Barrell, M.H. de Bruijn, A.R. Coulson, J. Drouin, I.C. Eperon, D.P. Nierlich, B.A. Roe, F. Sanger, P.H. Schreier, A.J. Smith, R. Staden, I.G. Young, Sequence and organization of the human mitochondrial genome, Nature 290 (1981) 457–465.
[4] G. Tully, W. Bär, B. Brinkmann, A. Carracedo, P. Gill, N. Morling, W. Parson, P. Schneider, Considerations by the European DNA profiling (EDNAP) group on the working practices, nomenclature and interpretation of mitochondrial DNA profiles, Forensic Sci. Int. 124 (2001) 83–91.
[5] M.R. Wilson, M.W. Allard, K.L. Monson, K.W. Miller, B. Budowle, Recommendations for consistent treatment of length variants in the human mitochondrial DNA control region, Forensic Sci. Int. 129 (2002) 35–42.
[6] H.J. Bandelt, W. Parson, Consistent treatment of length variants in the human mtDNA control region: a reappraisal, Int. J. Legal Med. 122 (2008) 11–12.
[7] W. Parson, A. Dür, EMPOP—a forensic mtDNA database, Forensic Sci. Int. Genet. 1 (2007) 88–92.
[8] D. Gusfield, Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press, New York, 1999.
[9] D.W. Mount, Bioinformatics: Sequence and Genome Analysis, Cold Spring Harbor Laboratory Press, New York, 2004.
[10] H.J. Bandelt, Q.P. Kong, M. Richards, V. Macaulay, Estimation of mutation rates and coalescence times: some caveats, in: H.J. Bandelt, V. Macaulay, M. Richards (Eds.), Human Mitochondrial DNA and the Evolution of Homo Sapiens, Springer-Verlag, Berlin, Germany, 2006, pp. 47–90.
[11] T.M. Diegoli, J.A. Irwin, R.S. Just, J.L. Saunier, J.E. O'Callaghan, T.J. Parsons, Mitochondrial control region sequences from an African American population sample, Forensic Sci. Int. Genet. 4 (2009) e45–e52.
[12] S. Lutz, H.J. Weisser, J. Heizmann, S. Pollak, Mitochondrial heteroplasmy among maternally related individuals, Int. J. Legal Med. 113 (2000) 155–161.
[13] M.R. Wilson, M.W. Allard, K.L. Monson, K.W.P. Miller, B. Budowle, Further discussion of the consistent treatment of length variants in the human mitochondrial DNA control region, Forensic Sci. Commun. 4 (2002) 4.
[14] A. Brandstätter, C.T. Peterson, J.A. Irwin, S. Mpoke, D.K. Köch, W. Parson, T.J. Parsons, Mitochondrial DNA control region sequences from Nairobi (Kenya): inferring phylogenetic parameters for the establishment of a forensic database, Int. J. Legal Med. 118 (2004) 294–306.
[15] A. Brandstätter, H. Niederstätter, M. Pavlic, P. Grubwieser, W. Parson, Generating population data for the EMPOP database - an overview of the mtDNA sequencing and data evaluation processes considering 273 Austrian control region sequences as example, Forensic Sci. Int. 166 (2007) 164–175.
[16] B. Zimmermann, M. Bodner, S. Amory, L. Fendt, A. Röck, D. Horst, B. Horst, T. Sanguansermsri, W. Parson, A. Brandstätter, Forensic and phylogeographic characterization of mtDNA lineages from northern Thailand (Chiang Mai), Int. J. Legal Med. 123 (2009) 495–501.
[17] M. van Oven, M. Kayser, Updated comprehensive phylogenetic tree of global human mitochondrial DNA variation, Hum. Mutat. 30 (2009) E386–E394.